

**PATENT ABSTRACTS OF JAPAN**(11)Publication number : **08-286958**(43)Date of publication of application : **01.11.1996**

(51)Int.Cl.

G06F 11/34

G06F 9/06

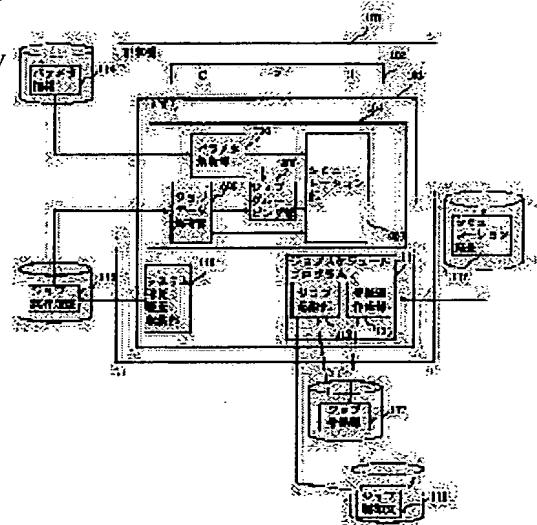
G06F 9/46

(21)Application number : **07-087966**(71)Applicant : **HITACHI LTD**(22)Date of filing : **13.04.1995**(72)Inventor : **HONDO YURI  
NAGASUGA HIROFUMI  
YAMAGISHI TADASHI  
TANAKA TOSHIHARU  
KINOSHITA TOSHIYUKI****(54) METHOD FOR ANALIZING JOB SCHEDULING**

(57)Abstract:

**PURPOSE:** To provide an effective job scheduling plan before a batch processing is executed and also to predict influence at the time of change in system configuration capable of parallelly executing plural jobs before change.

**CONSTITUTION:** The execution start conditions of respective jobs are decided based on a past job execution history 115 and the execution times of the respective jobs are obtained from execution history information. When change in a system operation such as job multiplexing increase/decrease is given as parameter information 114, the parameter is added as the condition and a simulation is executed in a simulation part 104 so that its result is provided as the job schedule plan.

**LEGAL STATUS**

[Date of request for examination] 16.02.2001

[Date of sending the examiner's decision of rejection] 12.04.2002

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-286958

(43)公開日 平成8年(1996)11月1日

(51)Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 11/34		7313-5B	G 0 6 F 11/34	S
9/06	4 1 0		9/06	4 1 0 Z
9/46	3 4 0		9/46	3 4 0 D

審査請求 未請求 請求項の数 5 O L (全 13 頁)

(21)出願番号 特願平7-87966

(22)出願日 平成7年(1995)4月13日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 本堂 友理

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72)発明者 長須賀 弘文

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72)発明者 山岸 正

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア開発本部内

(74)代理人 弁理士 磯村 雅俊

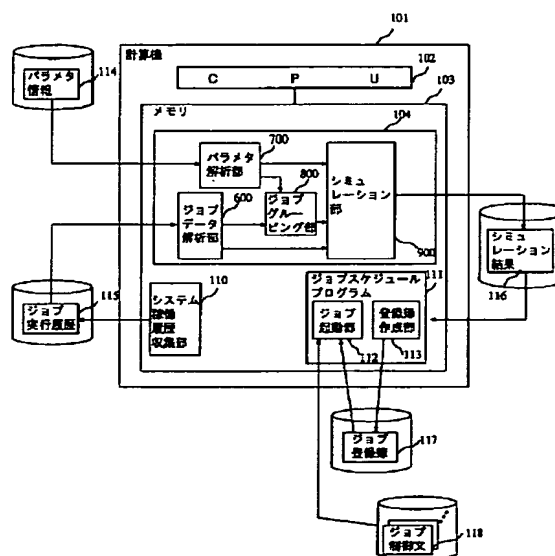
最終頁に続く

(54)【発明の名称】 ジョブスケジューリング解析方法

(57)【要約】

【目的】 複数のジョブが並行実行可能な計算機システムにおいて、バッチ処理を実施する前に、効率的なジョブスケジューリング案を提供するとともに、システム構成に変更があった場合の影響を変更前に予測すること。

【構成】 過去のジョブ実行履歴115をもとに各ジョブの実行開始条件を決定し、上記実行履歴情報から、上記各ジョブの実行時間を求める。ジョブ多重度増減などのシステム運用上の変更をパラメタ情報114として与えられた場合に、そのパラメタを条件として付加してシミュレーション部104でシミュレーションを行い、その結果をジョブスケジュール案として提供する。



【特許請求の範囲】

【請求項1】 複数のジョブが並行して動作することが可能な計算機システムにおけるジョブスケジューリング解析方法において、

動作した当該ジョブの実行履歴情報と、各ジョブが動作中に入出力操作を行ったファイルのアクセス履歴情報とを用いて各ジョブの実行のシミュレーションを行うことを特徴とするジョブスケジューリング解析方法。

【請求項2】 請求項1記載のジョブスケジューリング解析方法において、

上記各ジョブの実行のシミュレーションは、上記ジョブの実行履歴情報と上記アクセス履歴情報の外に、さらにシミュレーションの実行条件を決定するパラメタ情報を用いて行われるものであることを特徴とするジョブスケジューリング解析方法。

【請求項3】 複数のジョブが並行して動作することが可能な計算機システムにおけるジョブスケジューリング解析方法において、

動作した当該ジョブの実行履歴情報を収集するステップと、

各ジョブが動作中に入出力操作を行ったファイルのアクセス履歴情報を収集するステップと、

上記計算機システムで並行動作可能な最大ジョブ数（ジョブ多重度）を指定するステップと、

上記アクセス履歴情報から、ファイル受渡し関係が保たれるように、上記各ジョブの実行開始条件を決定するステップと、

上記実行履歴情報から、上記各ジョブの実行時間を求めるステップと、

並行して動作可能な最大ジョブ数が上記ジョブ多重度で、かつ、ジョブの実行順序は、上記実行開始条件を満たし、かつ、上記ジョブは上記実行時間で処理を行うように、上記各ジョブの実行のシミュレーションをするステップとを有することを特徴とするジョブスケジューリング解析方法。

【請求項4】 請求項3記載のジョブスケジューリング解析方法において、

計算機システムのハードウェア構成の変更内容を記述したパラメタ情報を読み込むステップをさらに有し、

上記各ジョブの実行のシミュレーションをするステップは、上記読み込んだ構成パラメタで指定された変更後のシステム構成の上でのジョブ実行をシミュレーションするものであることを特徴としたジョブスケジューリング解析方法。

【請求項5】 請求項4記載のジョブスケジューリング解析方法において、

上記ジョブ実行履歴情報と上記アクセス履歴情報から、実行中に同一のファイルにアクセスした上記各ジョブを同一グループに属するように、上記各ジョブを分類するステップをさらに有し、

上記分類されたグループ毎に、該グループに属するジョブの実行のシミュレーションを行うことを特徴としたジョブスケジューリング解析方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、複数のジョブが並行して動作することが可能な計算機システムにおけるジョブスケジューリング解析方法に関し、特に、構成の異なる計算機システムに移行する際に有効なジョブスケジューリング解析方法に関するものである。

【0002】

【従来の技術】従来、大規模なバンキングシステムにおけるオンライン処理や流通業における商品管理をするためのオンライン処理では、顧客サービスの向上を図るために、数多くの業務が実施されている。オンライン処理には、その業務で取り扱ったデータを集計するための一括データ処理（バッチ処理）が付随している。オンライン処理量の増加やオンラインサービスの種類の多様化に伴い、バッチ処理量は増大し、その処理内容も複雑化しつつある。一般に、オンライン処理を行うジョブをオンラインジョブ、バッチ処理を行うジョブをバッチジョブと呼ぶ。これらの増大化しつつある業務に対処するために、複数のCPUを用意した計算機システムの構築を行い、そのシステムでバッチ業務を実施する傾向にある。この複数CPUを有するシステム構築手法として、複数台のCPUを一つの主記憶装置を共有して一つのオペレーティングシステム（OS）で動作するシステム（密結合マルチプロセッサシステム）がある。もう一つのシステム構築手法として、独立したCPUおよび主記憶装置を持つ複数個の計算機を、チャネル間結合装置などで結合させたシステム（疎結合マルチプロセッサシステム）がある。

【0003】上述した両方の構築手法によるシステムにおいては、短時間でバッチ処理が完了するように効率のよいジョブスケジューリングを編成する必要がある。特に、疎結合マルチプロセッサシステムでは、複数の独立したOSが動作するため、OS間で相互に連携をとりながら業務を遂行する必要がある。そこで、バッチ処理を効率よく実施させる従来の手法として、特開平5-173807号公報や特開平6-68052号公報に開示されたものがある。この2件は、疎結合マルチプロセッサシステム上で、負荷が均等化されるように動的に資源の利用度を測定し、スケジューリングを行うジョブ分散方式に関するものである。すなわち、上記特開平5-173807号公報に記載されたものは、一定時間毎にシステムの資源負荷量を測定する資源負荷測定手段を設け、そこでの測定結果に応じてジョブ実行多重度を動的に変更させるものである。また、上記特開平6-68052号公報に記載されたものは、一定時間間隔でプロセッサの使用率を測定するプロセッサ使用率監視部を設け、最も負荷

の低いプロセッサにジョブを自動的に割り当てる機能に関するものである。

【0004】

【発明が解決しようとする課題】近年、業務の多様化やデータ量の増加に伴い、バッチ処理が複雑化していることは前述の通りである。ジョブの処理時間を短縮するための方法として、計算機のCPU性能を上げるかまたはジョブ全体を高多重化する手段が考えられる。しかしながら、前者の単一CPU性能を上げるには現在の技術ではほぼ限界が現れており、今後、後者のジョブの高多重化が必須になるのは確実と考えられる。しかし、これらの複雑なバッチジョブの相関関係を崩さずに、バッチジョブの高多重化を実現するには、ジョブのスケジューリングの効率化を行わなければ、マルチプロセッサシステムの計算機資源を有効に活用することにはならない。上記2つの公開公報に記載された従来技術は、既に構築されたシステム上でのジョブ実行中に動的にジョブスケジューリングを決定するものである。これは、ジョブが実行される際に、そのジョブ単体に対してスケジューリングの効率化を図るもので、ジョブ全体に対しての測定を行ってスケジューリングの効率化を行うものではなかった。また、これらの従来技術は、既に構築されたシステムでジョブを実行する場合に適用されるものであり、システム構成の変更を行う場合の前もったジョブスケジューリング評価に利用することができなかった。

【0005】オンラインジョブが時間に依存したジョブであるのに対し、一括データ処理であるバッチジョブは決まった入力ファイルに対し一定の処理を行う開始時刻に依存しないという性質を持つ。このような複数のジョブが定期的に繰り返されるような形態であるジョブに関しては、過去のジョブ実行履歴をもとに、その後のジョブ実行の予測が可能である。このような性質のジョブについて、実行される前にジョブスケジューリング計画を行うことによって、全体の処理時間の短縮を見込むことができる。また、システム設計時に前記のようなジョブスケジューリング計画を行うことで、ディスク装置などへのファイルの配置計画などを立てることができ、システム構成の変更を行う場合の前もったジョブスケジューリング評価を行うことができる。

【0006】本発明はバッチジョブのスケジューリング効率化を対象とし、以下の三つの目的を持つ。本発明の第一の目的は、バッチ処理を実施する前に、過去の実行履歴をもとに、効率的なジョブスケジュールの立案を支援することが可能なジョブスケジューリング解析方法を提供することにある。本発明の第二の目的は、ジョブ多重度などのシステム運用上の変更があった場合に、過去の実行履歴をもとに、その効果を変更前に見積もることが可能なジョブスケジューリング解析方法を提供することにある。本発明の第三の目的は、疎結合マルチプロセ

ム内の他の計算機に接続されているディスクへのアクセス頻度を少なくし、データ転送に関するオーバーヘッドを低くおさえるため、分配されるジョブがアクセスするファイルが上記システム内の一つの計算機に接続されているディスクに配置することが可能となるように、各ジョブを実行させる計算機を、過去の履歴情報をもとに決定することが可能なジョブスケジューリング解析方法を提供することにある。

【0007】

【課題を解決するための手段】本発明は、上記第一の目的を達するために、複数のジョブが並行して動作することが可能な計算機システム上で動作した各ジョブの実行履歴情報と当該各ジョブが動作中に入出力操作を行ったファイルのアクセス履歴情報を用いてシミュレーションを行なう。そのための手法として、各バッチジョブは、参照するファイルが生成されていれば、論理的に実行可能であることに着目して、上記アクセス履歴情報から、各ジョブの実行開始条件を決定するステップと、上記実行履歴情報から、上記各ジョブの実行時間を求めるステップとを設ける。そして、シミュレーションの対象である計算機システムにおけるジョブ多重度を、並行して動作可能な最大ジョブ数とする第一条件と、各ジョブの実行順序は、上記実行開始条件に従うという第二条件をもとにしてシミュレーションを行うステップを設ける。また本発明は、上記第二の目的を達するために、ジョブ多重度増減などのシステム運用上の変更をパラメタとして与えられた場合に、該パラメタを第三条件として付加するステップをさらに設けてシミュレーションを行なう。さらに、本発明は、上記第三の目的を達するために、上記シミュレーションを行なう前に、予めアクセスするファイルの局所性に基づいてジョブのグルーピングを行うステップを設ける。

【0008】

【作用】本発明によると、収集したアクセス履歴情報から各ジョブの実行開始条件を決定し、実行履歴情報から各ジョブの実行時間を求めることにより現状のジョブ実行に基づいたジョブ実行開始時間を求めることができる。そして、シミュレーションの対象である計算機システムにおけるジョブ多重度を、並行して動作可能な最大ジョブ数とする第一条件と、上記実行開始条件に従うという第二条件をもとにしてシミュレーションを行うステップにより、ジョブスケジュールによらない仮想的なジョブ実行を再現できる。このシミュレーション結果から、処理時間短縮がはかれるジョブスケジュール案を提示することができる。

【0009】ジョブ多重度増減などのシステム運用上の変更をパラメタとして与えられた場合に、そのパラメタを上記実行開始条件などに付加することにより、運用上の変更を行った場合のジョブスケジュール結果を提供できる。その結果、現状システムでの問題点を解決する見

積りを得た上でのシステム再構築が可能となる。また、シミュレーションを行う前に、あらかじめアクセスするファイルに基づいてジョブのグルーピングを行うことにより、パラメタとして与えられた新しいシステム条件でのジョブスケジュール結果を提供できる。その結果、疎結合マルチプロセサシステムなどの現行のシステムと違った構成のシステム上でのジョブスケジュール案を提示することができる。以上のように、実行履歴を用いる静的な評価により、本発明は実機を用いることなく、すなわち対象となるシステムに負担をかけることなくジョブスケジュールに関する定量的評価を得ることが可能になる。

#### 【0010】

【実施例】本実施例では、一つ以上のCPU、および、そのCPUからアクセスされるディスクを有する一つ以上の計算機により構成される計算機システムを対象としている。計算機システムが複数の計算機から構成される場合には、それら各計算機は相互に通信を行う手段を有するものとする。上記の計算機システムにおいて、複数のジョブは並行実行される。ここで、並行実行されるジョブの数には上限があり、その上限をジョブ多重度と呼ぶ。ジョブは、一つ以上のジョブステップにより構成される。ここでジョブステップは、そのジョブの中で実行されるプログラムの一つに対応している。read、write、参照などの動作が行われたファイルを、本件ではアクセスファイルと呼ぶ。バッチジョブは、参照するファイルが生成されていれば論理的に実行可能である。従って、あるジョブがアクセスするファイルが他のジョブによってアクセスされている場合には、その他のジョブが終了した後に実行開始する必要がある。ジョブが、どのジョブの実行終了後に開始すべきであるかをスケジューリング条件と呼ぶ。

【0011】以下、本発明の実施例を図面を用いて詳細に説明する。図1は、本発明の一実施例の基本的なシステム構成を示している。本実施例において、計算機101は、CPU102、メモリ103により構成されている。メモリ103上には、後述するパラメタ情報114およびジョブ実行履歴115を入力してジョブスケジューリング解析を行なうジョブスケジューリング解析部104、システムの稼働履歴を収集するシステム稼働履歴収集部110、およびジョブスケジュールプログラム111が設けられ、さらに、ジョブスケジュールプログラム111はジョブ起動部112および登録作成部113から構成されている。これらの各部における処理はCPU102によって実行されるものとする。

【0012】計算機101に登録されたジョブは、ジョブスケジュールプログラム111内の登録簿作成部113により、ジョブ登録簿117に登録される。ジョブ起動部112は、上記の手順により作成されたジョブ登録簿117をもとに、ジョブを起動し実行状態にする。起

動するには、起動しようとするジョブが、どのプログラムを実行するもので、どのファイルにアクセスするかを記述したジョブ制御文118が各ジョブ毎に用意されている必要がある。ジョブ起動部112は、このジョブ制御文118をもとに、ジョブを実行するのに必要なファイルなどの資源が確保されているかを確認した上でジョブを起動する。

【0013】ジョブは上記手順により実行状態になる。その実行状況をシステム稼働履歴収集部110が収集してジョブ実行履歴115を作成する。このジョブ実行履歴115の構成例を図2に示す。採取する実行履歴データとしては、同図に示すように、ジョブ開始レコード115-1、ジョブステップ開始レコード115-2、アクセスファイルに関するファイルクローズレコード115-3、ジョブステップ終了レコード115-4、ジョブ終了レコード115-5などが挙げられる。これらの各レコードは、ジョブの実行開始、ファイルのクローズなどの事象によって発生するものであり、それらのレコードの先頭または内部にそのレコードが何の事象によって発生したかを表すレコードタイプが含まれている。本実施例では、ジョブ開始レコードを01、ジョブステップ開始レコードを02、ファイルクローズレコードを03、ジョブステップ終了レコードを04、ジョブ終了レコードを05とするレコードタイプに関する記述が、各レコードの先頭にあるものとする。

【0014】図2に示されたジョブ実行履歴データである各レコードについてさらに詳細に説明する。ジョブ開始レコード115-1は、レコードタイプ01の他に、ジョブ名、ジョブ開始日付、ジョブ開始時刻を有し、ジョブステップ開始レコード115-2は、レコードタイプ12の他に、ジョブステップ名、ジョブステップ開始日付、ジョブステップ開始時刻、このレコードを出したジョブ名を有し、アクセスファイルに関するファイルクローズレコード115-3は、レコードタイプ03の他に、ファイル名、ファイルクローズ日付、ファイルクローズ時刻、このレコードを出したジョブステップ名、このレコードを出したジョブ名を有し、ジョブステップ終了レコード115-4は、レコードタイプ04の他に、ジョブステップ名、ジョブステップ終了日付、ジョブステップ終了時刻、このレコードを出したジョブ名を有し、ジョブ終了レコード115-5は、レコードタイプ05の他に、ジョブ名、ジョブ終了日付、ジョブ終了時刻、CPU使用時間を有している。

【0015】ジョブ実行履歴115とともにジョブスケジューリング解析部104へ入力されるパラメタ情報114の構造を図3に示す。パラメタ情報114は、ジョブに関するデータである図2に詳細に示したジョブ実行履歴115以外の、シミュレーションに必要な、空間数（最大ジョブ実行多重度）や、CPU性能比率（新たなシステムにおけるCPUの性能の改善比率）や、出力結

果の指定などの全ての情報を含む。図3は空間数30、CPU性能比率1（同一性能のCPUの場合）、グルーピングON（グルーピング機能を有効にした場合）、・・・の場合を示している。

【0016】次に、本発明が特徴としているジョブスケジューリング解析部104の詳細を説明する。ジョブスケジューリング解析部104は、図1に示したように、上述した実行履歴データ115（図2参照）を解析するジョブデータ解析部600、上述したパラメタ情報114（図3参照）からシミュレーションに必要な情報であるシミュレーション条件定義情報（図5の114-1）を作成するパラメタ解析部700、ジョブデータ解析部600により作成されたデータについてジョブのグルーピングを行うジョブグルーピング部800、これらの入力データによりジョブの実行状況をシミュレートするシミュレーション部900により構成される。ジョブスケジューリング解析部104と、システム稼働履歴収集部110は、本実施例では同一システム内に設けているが、別システムに設けてもよい。

【0017】また、シミュレーション部900で解析された結果が、シミュレーション結果116として、シミュレーション部900より出力される。このシミュレーション結果116の例を図4に示す。出力情報としては、同図に示すように、シミュレーション後のジョブ実行多重度の時間的遷移116-1や、実行ジョブのリスト116-2などが挙げられる。図4の実行ジョブのリスト116-2は、空間1の実行ジョブはJOB1、JOB5、およびJOB3で、空間2の実行ジョブはJOB2およびJOB8であり、例えば、JOB1は7:00乃至7:45に実行され、実行時間は45分間であることを示している。なお、実行ジョブのリスト中にジョブ毎に使用される資源などを含めることも可能である。

【0018】次に、ジョブスケジューリング解析部104の詳細な説明図を図5に示す。ジョブデータ解析部600は、ジョブ実行履歴115から本発明でシミュレーションに用いる情報単位である情報セル200を作成する。情報セル200は、ジョブ実行履歴115の中に含まれる履歴の種類によって、図6の情報セル構成例に示したように、ジョブ情報セル201、ジョブステップ情報セル202、ファイル情報セル203に大別される。情報セル200はテーブル構造になっており、その中の要素（ポインタ）が関連のある情報セルをポイントすることで、情報セル全体がリスト構造になるように構成されている。ジョブ情報セル201は、そのジョブを構成するジョブステップ情報セル202のリストを指す。また、そのジョブステップ情報セル202は、そのジョブステップでアクセスするファイルのファイル情報セル203を指す。これらが1組となって、ジョブの全体の情報を表す。ジョブ実行履歴115には解析の対象となるジョブが複数含まれているため、ジョブ情報セル201

自身も、リスト構造となっており、ポインタによって順次次のジョブ情報セル201を指すように構成されている。

【0019】次に、上記の各情報セルのさらに詳細なテーブル構造を図7～図9を用いて説明する。各情報セルには、情報セルのリスト構造を形成するための情報とともに、実行履歴の中に含まれる情報が必要に応じて格納される。図7はジョブ情報セル201のテーブル構造を示すものである。同図中、ポインタ301はジョブ情報セルのリストにおいて次の要素をポイントするもので、ポインタ302は、そのジョブを構成するジョブステップの情報セルのリストの先頭をポイントするものである。また、303はジョブを識別するためのジョブ名を、304はジョブ開始時刻を、305はジョブ終了時刻を格納する領域であり、これらはシミュレーション部900で実行条件を求める際に使用される。また、306はそのジョブが使用したトータルのCPU使用時間を格納する領域である。

【0020】図8はジョブステップ情報セル202のテーブル構造を示すものである。同図中、ポインタ401はジョブステップ情報セルのリストにおいて次の要素（ジョブステップ情報セル）をポイントするもので、ポインタ402は、そのジョブステップでアクセスするファイルの情報セルのリストの先頭をポイントするものである。403はジョブステップを識別するためのジョブステップ名を、404はそのジョブステップで実行されるプログラム名を格納するための領域である。図9はファイル情報セル203のテーブル構造を示すものである。同図中、ポインタ501はファイル情報セルのリストにおいて次の要素（ファイル情報セル）をポイントするためのものである。502はファイルを識別するとともに、シミュレーション部900で実行条件を求める際に使用されるファイル名を格納するための領域である。

【0021】次に、ジョブデータ解析部600の処理の流れを図10のフローチャートを用いて詳細に説明する。まず、ジョブ実行履歴115から、先頭のデータを読み込み、ステップ602乃至ステップ606の処理をジョブ実行履歴115のデータが空になるまで継続する（ステップ601）。そのレコードが読み込まれるまでの情報セルリストの中に、そのレコードの内容を格納すべき情報セルが存在しないレコード、例えば、ジョブ開始レコードなどを読み込んだ場合、新たに内容を格納する情報セルを作成する必要がある。読み込んだジョブ実行履歴115のデータが、これらの新たに情報セルを作成すべきレコードタイプであるか否かを判定する（ステップ602）。

【0022】上記ステップ602における判定が真であった場合、該レコードがジョブか、ジョブステップか、またはファイルに関するものであるかをそのレコードの内容から判定し、それに該当する形式の情報セルの領域

を確保し(ステップ603)、その領域にデータを格納するとともに、その新しい情報セルに対するリストからのポインタ設定など、リスト構造を形成するための情報を設定する(ステップ604)。上記ステップ602における判定が偽であった場合、それ以前に作成された情報セルの中の対応する情報セルにデータを格納する(ステップ605)。その後、ジョブ実行履歴115の次のデータの処理を対象として判定ステップ601に戻る(ステップ606)。

【0023】パラメタ解析部700は、図3に示した如きパラメタ情報114を解析して、シミュレーションの条件となるシミュレーション条件定義情報114-1を作成するものである。次に、パラメタ解析部700の処理を図11のフローチャートを用いて詳細に説明する。まず、パラメタ情報114からパラメタの読み込みを行い、ステップ702乃至ステップ703に示した処理をパラメタ情報114のパラメタが空になるまで継続する(ステップ701)。読み込んだパラメタ情報がどのパラメタであるかを判別し、パラメタのタイプに合わせてシミュレーション条件定義情報114-1に反映させる(ステップ702)。例を挙げると、パラメタ情報114から、ジョブが実行される多重度を空間数として読み込み、シミュレーション条件定義情報114-1の中の対応する項目に設定する。処理を継続するために次のパラメタ情報114を対象として判定ステップ701に戻る(ステップ703)。

【0024】ジョブグルーピング部800は、現行のシステムと異なる疎結合マルチプロセッサシステムを意識した、ファイルアクセスの局所性に着目したジョブグルーピングを行った結果を得る場合に用いられるものである。ジョブグルーピング部800の処理により、ジョブ情報セル201のリストを、アクセスするファイルが重複するジョブの情報セルのリストに分割しグルーピングする。該ジョブグルーピング部800で作成されたジョブグループ単位でシミュレーション部900によりシミュレーションを行うことで、アクセスするファイルの重複をなくした、あるいは少なくしたジョブグループでのスケジューリング解析結果を出すことが可能になる。

【0025】次に、ジョブグルーピング部800の処理を図12のフローチャートを用いて詳細に説明する。まず最初に、シミュレーション条件定義情報114-1の中のジョブのグルーピングを行うか否かを指定しているテーブルデータを参照し、グルーピングを行うと指定してある場合に以下のステップ802乃至ステップ804の処理を行い、それ以外は処理を行わないで処理を終了する(ステップ801)。

【0026】ステップ801において、グルーピングを行なうと指定してある場合には、まず、ジョブ情報セル201の読み込み動作を行い、以下のステップ802乃至ステップ804の処理をジョブ情報セル201のリス

トが空になるまで継続する(ステップ802)。ステップ802で読み込んだジョブ情報セルで表されるジョブと同一ファイルにアクセスするジョブを、ジョブ情報セルのリストの中から順次検索し、これらのジョブを同一グループとみなしてグルーピングする。また、ここで同一グループとされたジョブと同一ファイルにアクセスするジョブも同様に同一グループとする(ステップ803)。次のジョブ情報セルを対象として(ステップ804)、ステップ802に戻って再び同様の処理を行なう。

【0027】次に、シミュレーション部900におけるジョブ実行のシミュレーションの手順を図13および図14のフローチャートを用いて詳細に説明する。図14は、図13で行われるジョブ情報セル201のリストつなぎ換えを模式的に表したものである。図14において、201-j(j=1~i)は内容の異なるジョブ情報セルを表す。図14中の空間1001のテーブル構造例を図15に示す。図15に示した空間1001のテーブル構造において、ポインタ1101は次の空間1001をポイントするためのものである。また、ポインタ1102はその空間1001に接続されるジョブ情報セルのリストの先頭をポイントするためのものである。

【0028】次に、図13のフローチャートに沿ってシミュレーション部900の動作を詳細に説明する。まず最初に、シミュレーション条件定義情報114-1から情報を読み込む。シミュレーション条件定義情報114-1が空になるまで以下のステップ902乃至ステップ903の処理を行う(ステップ901)。シミュレーション条件定義情報114-1が空になったら、ステップ904に進む。

【0029】ステップ902では、読み込んだシミュレーション条件に対応した処理を行う。例えば、ジョブ実行多重度の最大数を表す空間数のパラメタであった場合は、その数の空間1001をメモリ領域に確保する。またCPU性能比率であった場合は、ジョブ情報セル201内のCPU使用時間306を該比率に合わせて換算し、同時にジョブ終了時刻305の書き換え(図7参照)も行う。

【0030】接続するジョブ情報セルのリスト1の先頭から、ジョブ情報セル201を順々に取り出すために、ジョブ情報セル201があるかどうかの判定を行い(ステップ904)、ジョブ情報セル201があった場合、先頭からそれを順に取り出し空になるまで以下のステップ905乃至ステップ911の処理を行う。まず、上記のジョブの、他のジョブとのファイル受渡しなどのスケジューリング条件を求める(ステップ905)。次に、ステップ905で求めたスケジューリング条件により実行開始可能な時刻を算出する(ステップ906)。ステップ906で求めた時刻に空間1001に接続されたジョブ情報セルリストの内容を検索し、上記実行開始可能

な時刻に空きがあり、実行可能であるか否かを判定する(ステップ907)。

【0031】上記ステップ907における判定が真(実行可能)であった場合、その空いている空間1001にジョブ情報セルを接続する。また、ジョブ情報セルを接続した際、そのジョブ情報セルが指し示すジョブステップ情報セルなども一緒に空間1001に接続する(ステップ908)。上記ステップ907における判定が偽(実行不可)であった場合、上記ステップ906で求めた実行開始可能時刻以降で、最も早く実行が可能となる空間1001に、ジョブ情報セルを接続する(ステップ909)。その後、上述したジョブ情報セルの接続が終了した後、ジョブ実行時間などのジョブ情報セル内のデータの書換えを行い(ステップ910)、次のジョブ情報セルを対象として判定ステップ904に戻る。なお、シミュレーション部900での処理中に、ジョブが実行されない時間が生じることがある。これらの時間を空きであることを表すジョブ情報セルで表すことで、空間1001のジョブ情報セルリスト内でのアイドル時間を表現することができる。

【0032】次に、シミュレーション部900におけるステップ904乃至ステップ911によって具体的にどのようにジョブ情報セルが変更されるかを図14を参照しながら説明する。まず、判定ステップ904において、接続すべき未処理のジョブ情報セルのリストを参照する。図14に示した例の場合、未処理のジョブ情報セルとして201-1、201-2、…、201-iがあるから判定ステップ904はYESであるため、リストの先頭から201-1、201-2、…、201-iを順に取り出してそれぞれに対してステップ905乃至ステップ911の処理を行なう。

【0033】最初に、リストの先頭のジョブ情報セル201-1を処理するにあたり、ステップ905乃至ステップ908またはステップ909において、シミュレーションの結果をメモリ上で表したジョブ情報セルのリストの中でジョブ情報セル201-1がどのジョブの後で実行されるべきかを割出す。図14における例の場合、ジョブ情報セル201-4の後に実行されるものとする。その場合、ステップ908またはステップ909において、ジョブ情報セル201-1は201-4の後につなぎ換えられる。同時にジョブ情報セル201-1は未処理のジョブ情報セルリスト200からは削除される。以上説明したように、本実施例では、ジョブの実行状況をメモリ上で情報セルのリストとして表し、パラメタ情報とジョブ実行履歴を用いてシミュレーションを行う。

【0034】

【発明の効果】本発明によれば、現行のシステムでのジョブ処理時間短縮が図れるジョブスケジューリング案を得られる。さらに、システム構成変更によるジョブ実行

状況が、システム構成変更前に、現行システムのジョブ稼働状況をもとにして把握することが可能になる。さらに、ジョブ実行履歴に基づいてシミュレーションを行うため、対象となるシステムに動的なジョブモニタリング程の負担をかけずに静的に評価を行うことが可能となる。

【図面の簡単な説明】

【図1】本発明のジョブスケジューリング解析方法を実現するためのジョブスケジューリング解析部104を具備した計算機システムの一実施例である。

【図2】本発明の入力情報となるジョブ実行履歴115の説明図である。

【図3】本発明の入力情報となるパラメタ情報114の説明図である。

【図4】本発明の出力情報となるシミュレーション結果116の説明図である。

【図5】本発明のジョブスケジューリング解析部104の具体的な構成例である。

【図6】本発明のジョブスケジューリング解析方法で用いられるジョブおよびジョブに関するデータである情報セルの構成を示した説明図である。

【図7】ジョブのデータを格納するジョブ情報セルのテーブル構造を示した説明図である。

【図8】ジョブステップのデータを格納するジョブステップ情報セルのテーブル構造を示した説明図である。

【図9】アクセスファイルのデータを格納するファイル情報セルのテーブル構造を示した説明図である。

【図10】ジョブデータ解析部600の処理を説明したフローチャートである。

【図11】パラメタ解析部700の処理を説明したフローチャートである。

【図12】ジョブグルーピング部800の処理を説明したフローチャートである。

【図13】スケジューリングのシミュレーション部900の処理を説明したフローチャートである。

【図14】シミュレーションでのジョブ実行を表すためのデータ構造を示した説明図である。

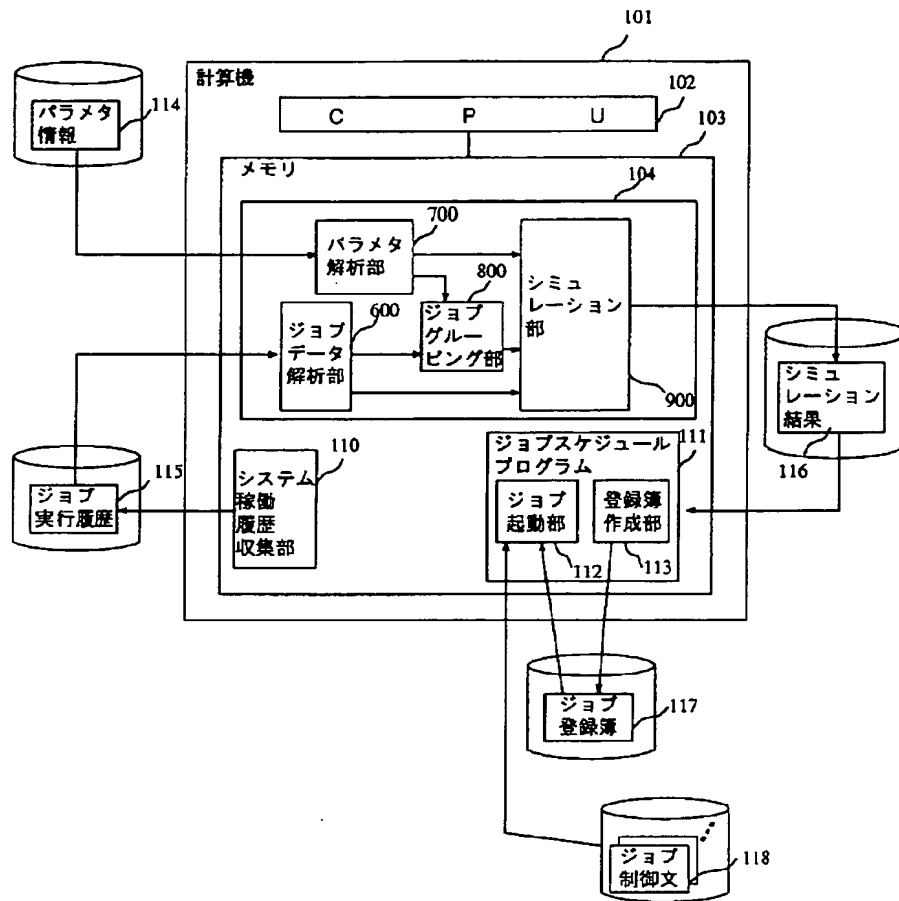
【図15】シミュレーションで用いられる空間1001のテーブル構造を示した説明図である。

【符号の説明】

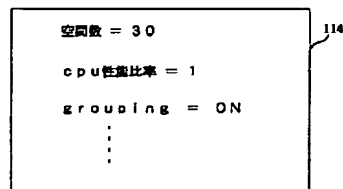
101: 計算機、102: CPU、103: メモリ、104: ジョブスケジューリング解析部、600: ジョブデータ解析部、700: パラメタ解析部、800: ジョブグルーピング部、900: シミュレーション部、110: システム稼働履歴収集部、111: ジョブスケジュールプログラム、112: ジョブ起動部、113: 登録簿作成部、114: パラメタ情報、115: ジョブ実行履歴、116: シミュレーション結果、117: ジョブ登録簿、118: ジョブ制御文



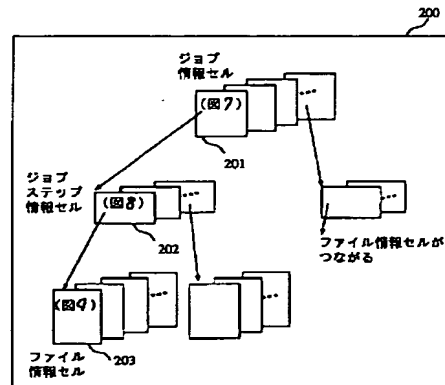
【図 1】



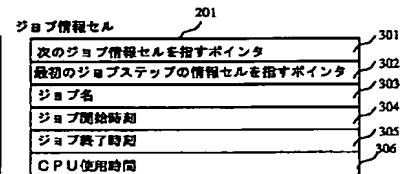
【図 3】



【図 6】



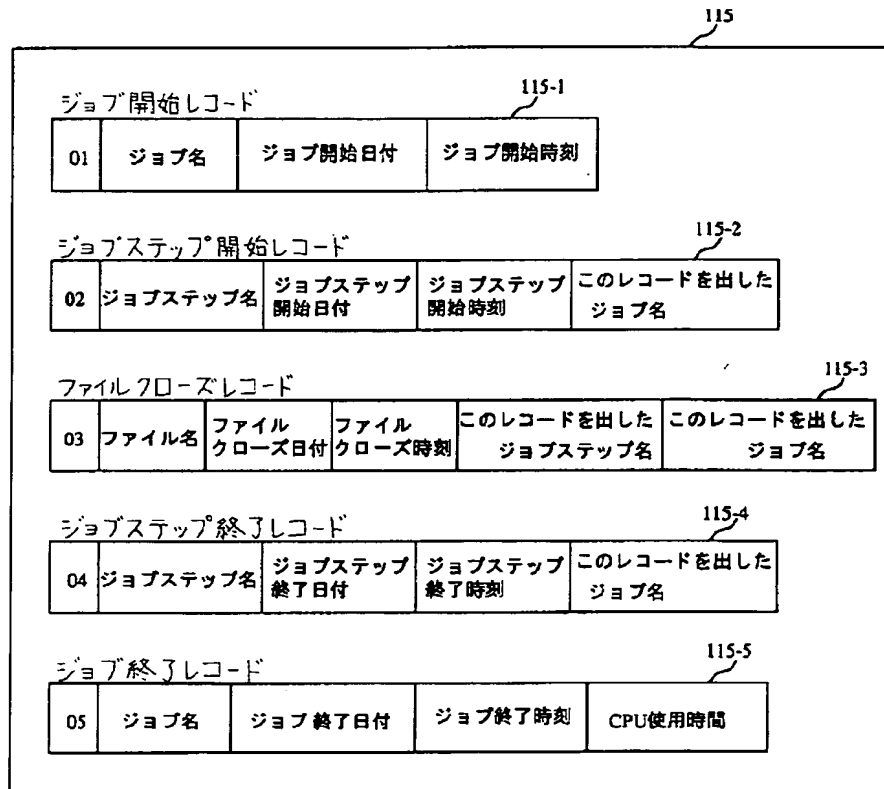
【図 7】



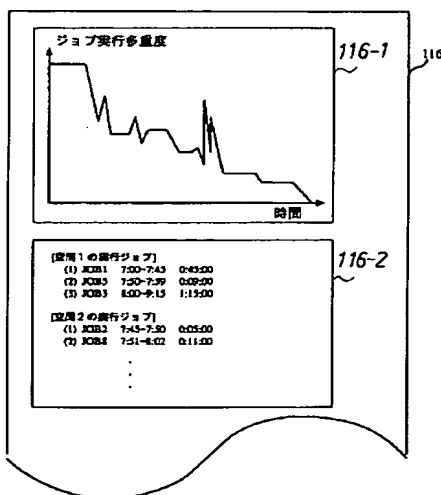
【図 9】



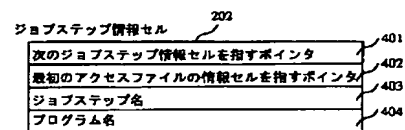
【図2】



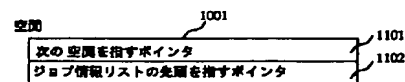
【図4】



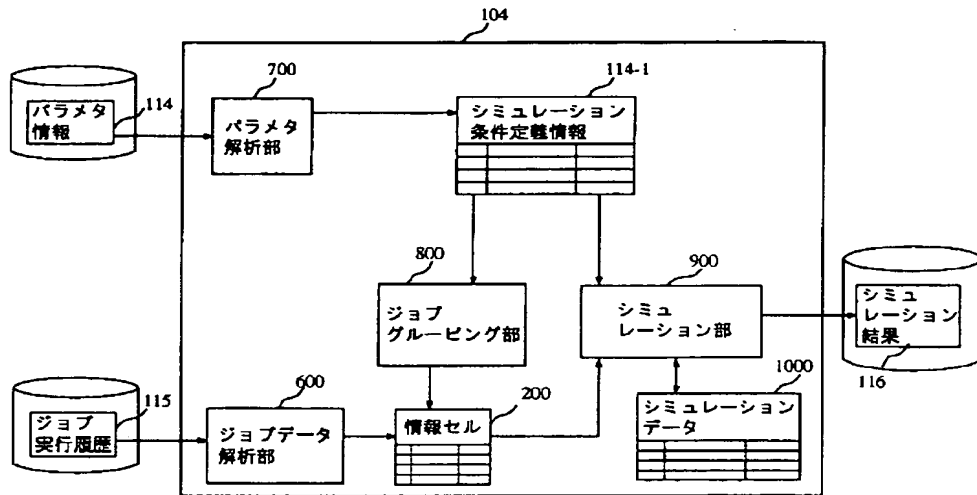
【図8】



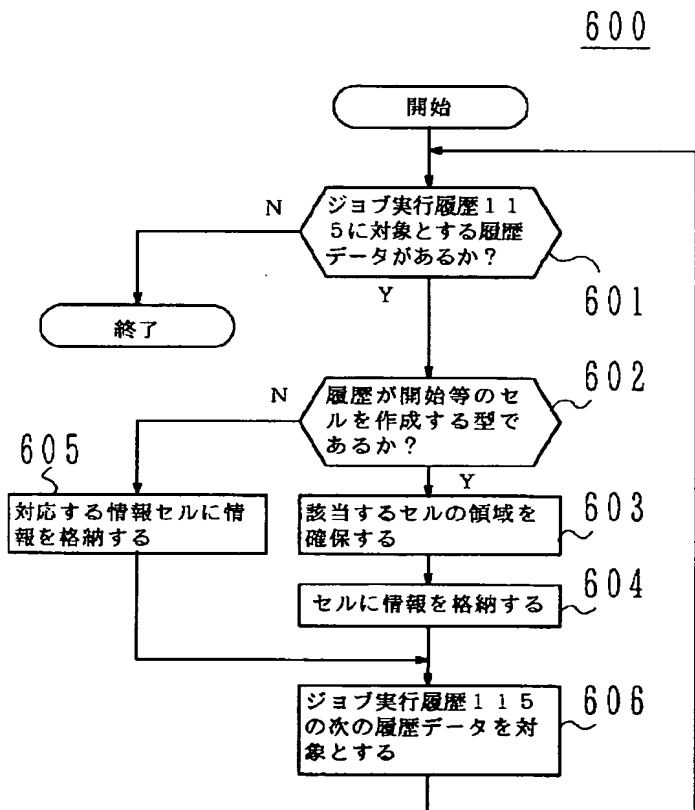
【図15】



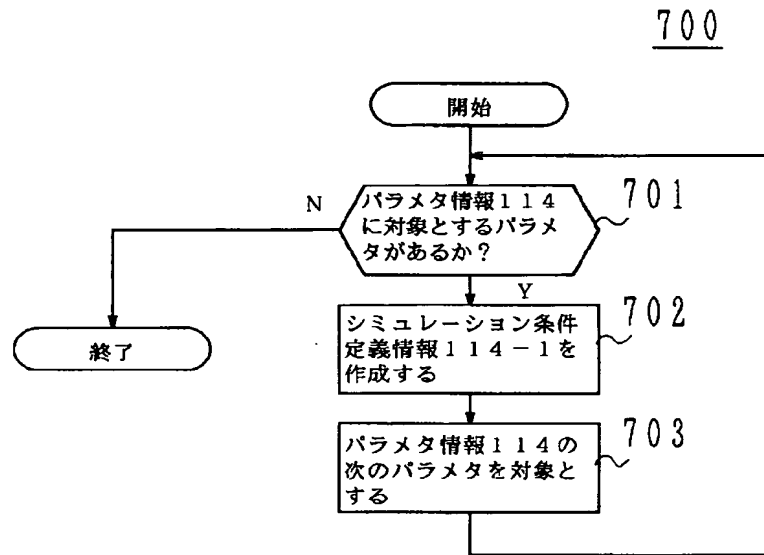
【図 5】



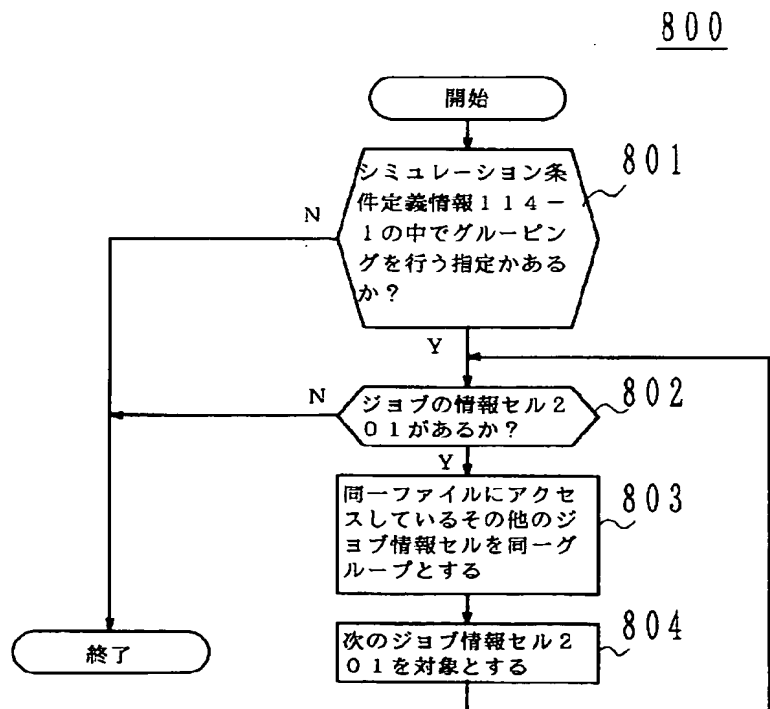
【図 10】



【図11】

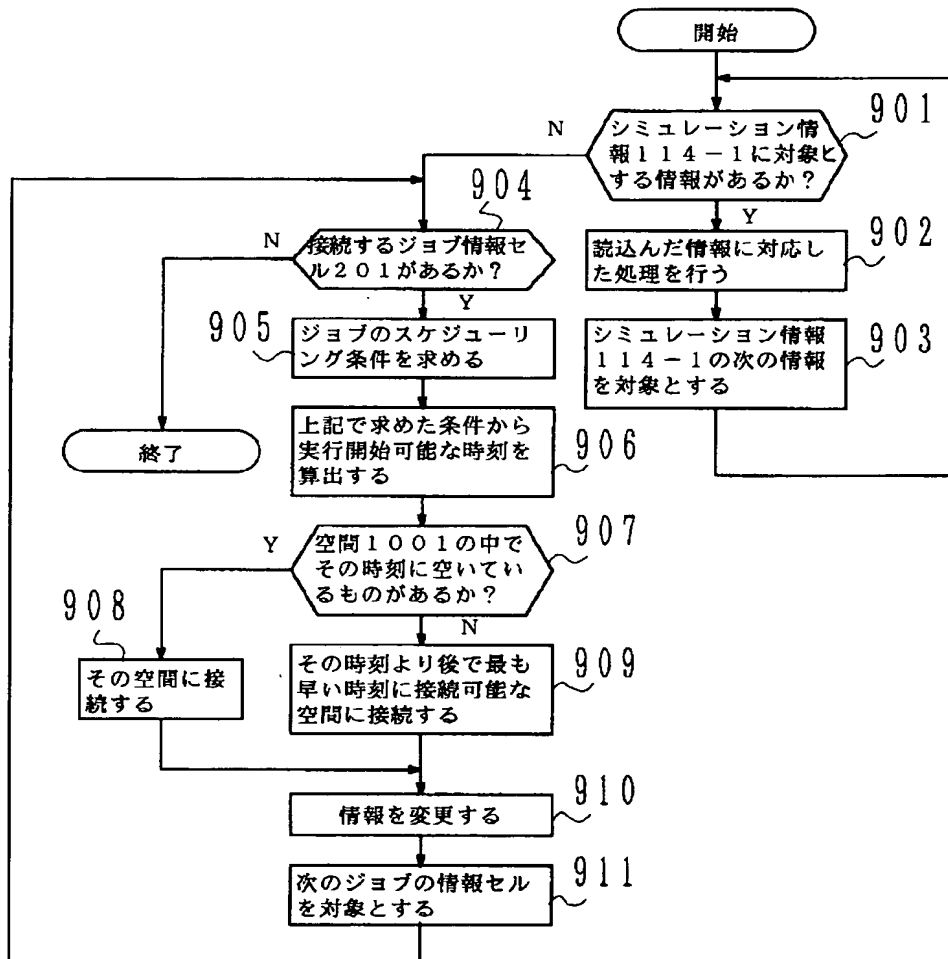


【図12】

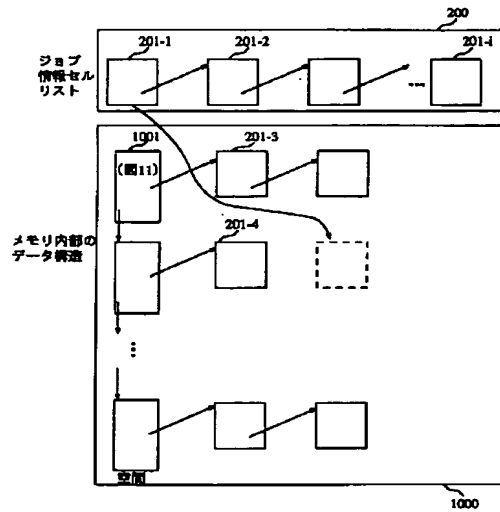


【図13】

900



【図 1 4】




---

フロントページの続き

(72)発明者 田中 俊治  
 神奈川県川崎市麻生区王禅寺1099番地 株  
 式会社日立製作所システム開発研究所内

(72)発明者 木下 俊之  
 神奈川県川崎市麻生区王禅寺1099番地 株  
 式会社日立製作所システム開発研究所内

【公報種別】特許法第17条の2の規定による補正の掲載  
 【部門区分】第6部門第3区分  
 【発行日】平成13年11月9日(2001.11.9)

【公開番号】特開平8-286958  
 【公開日】平成8年11月1日(1996.11.1)  
 【年通号数】公開特許公報8-2870  
 【出願番号】特願平7-87966  
 【国際特許分類第7版】

G06F 11/34  
       9/06 410  
       9/46 340

【FI】

G06F 11/34 S  
       9/06 410 Z  
       9/46 340 D

【手続補正書】

【提出日】平成13年2月16日(2001.2.16)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 複数のジョブが並行して動作することが可能な計算機システムにおけるジョブスケジューリング解析方法において、動作した当該ジョブの実行履歴情報と、各ジョブが動作中に入出力操作を行ったファイルのアクセス履歴情報とを用いて各ジョブの実行のシミュレーションを行うことを特徴とするジョブスケジューリング解析方法。

【請求項2】 請求項1記載のジョブスケジューリング解析方法において、上記各ジョブの実行のシミュレーションは、上記ジョブの実行履歴情報と上記アクセス履歴情報の外に、さらにシミュレーションの実行条件を決定するパラメタ情報を用いて行われるものであることを特徴とするジョブスケジューリング解析方法。

【請求項3】 請求項1または2記載のジョブスケジューリング解析方法において、

上記ジョブの実行履歴情報と上記アクセス履歴情報から、実行中に同一のファイルにアクセスした上記各ジョブを同一グループに属するように、上記各ジョブを分類するステップをさらに有し、

上記分類されたグループ毎に、該グループに属するジョブの実行のシミュレーションを行うことを特徴とするジョブスケジューリング解析方法。

【請求項4】 請求項1～3のいずれかに記載のジョブ

スケジューリング解析方法において、

上記シミュレーション結果を出力するステップとをさらに有することを特徴とするジョブスケジューリング解析方法。

【請求項5】 複数のジョブが並行して動作することが可能な計算機システムにおけるジョブスケジューリング解析方法において、動作した当該ジョブの実行履歴情報を収集するステップと、

各ジョブが動作中に入出力操作を行ったファイルのアクセス履歴情報を収集するステップと、

上記計算機システムで並行動作可能な最大ジョブ数(ジョブ多重度)を指定するステップと、

上記アクセス履歴情報から、ファイル受渡し関係が保たれるように、上記各ジョブの実行開始条件を決定するステップと、

上記実行履歴情報から、上記各ジョブの実行時間を求めるステップと、

並行して動作可能な最大ジョブ数が上記ジョブ多重度で、かつ、ジョブの実行順序は、上記実行開始条件を満たし、かつ、上記ジョブは上記実行時間で処理を行うように、上記各ジョブの実行のシミュレーションをするステップとを有することを特徴とするジョブスケジューリング解析方法。

【請求項6】 請求項5記載のジョブスケジューリング解析方法において、

計算機システム構成の変更内容を記述したパラメタ情報を読み込むステップをさらに有し、

上記各ジョブの実行開始条件を決定するステップは、上記読み込んだパラメタ情報で指定された変更後のシステム構成の上での条件であり、上記各ジョブの実行のシミュレーションをするステップは、上記読み込んだ構成バ

ラメタで指定された変更後のシステム構成の上でのジョブ実行をシミュレーションするものであることを特徴とするジョブスケジューリング解析方法。

【請求項7】 請求項5または6記載のジョブスケジューリング解析方法において、

上記ジョブの実行履歴情報と上記アクセス履歴情報から、実行中に同一のファイルにアクセスした上記各ジョブを同一グループに属するように、上記各ジョブを分類するステップをさらに有し、

上記分類されたグループ毎に、該グループに属するジョブの実行のシミュレーションを行うことを特徴とするジョブスケジューリング解析方法。

【請求項8】 請求項5～7のいずれかに記載のジョブスケジューリング解析方法において、

上記シミュレーション結果を出力するステップとをさらに有することを特徴としたジョブスケジューリング解析方法。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0015

【補正方法】変更

【補正内容】

【0015】ジョブ実行履歴115とともにジョブスケジューリング解析部104へ入力されるパラメタ情報114の構造を図3に示す。パラメタ情報114は、ジョブに関するデータである図2に詳細に示したジョブ実行履歴115以外の、シミュレーションに必要な、空間数（最大ジョブ実行多重度）や、CPU性能比率（新たなシステムにおけるCPUの性能の改善比率）や、出力結果の指定などの全ての情報を含む。図3は空間数30、CPU性能比率1（同一性能のCPUの場合）、グルーピングON（グルーピング機能を有効にした場合）の場合を示している。